

MULTIMEDIA



UNIVERSITY

STUDENT ID NO

--	--	--	--	--	--	--	--	--	--

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2015/2016

ECP4136 – JAVA TECHNOLOGY
(All Sections / Groups)

10 OCTOBER 2015
2:30 P.M – 4:30 P.M.
(2 Hours)- OPEN BOOK

INSTRUCTIONS TO STUDENT

1. This examination paper consists of FOUR (4) pages including cover with THREE (3) questions only.
2. Answer ALL questions. The distribution of the marks for each question is given.
3. This is an Open-Book examination.
4. Please write all your answers in the Answer Booklet provided.

Question 1

Define a **MeanCalculator** class with the following specifications:

- It implements mean calculations by defining these “static” method:
`+getArithmeticMean(numbers:double[]):double` method [2 marks]
`+getGeometricMean(numbers:double[]):double` method [2 marks]
`+getQuadraticMean(numbers:double[]):double` method [2 marks]
 Note: These methods return the means of **numbers** using the following formula.

Arithmetic mean:	$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{1}{n} (x_1 + \dots + x_n)$
Geometric mean:	$\left(\prod_{i=1}^n x_i \right)^{1/n} = \sqrt[n]{x_1 \cdot x_2 \cdot \dots \cdot x_n}$
Quadratic mean:	$\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}}$

- It has a static `+getRandom(from:double, to:double, arraySize:int, floatingPoint:int):double[]` method that returns an array of double values using parameters given. For example calling `getRandom(1.1, 1.2, 3, 2)` might return {1.11, 1.19, 1.13}. Hint: Generate integer in proper range and then divide it with 10^n , where n is 2 in the previous example. [5 marks]
- It has a driver function that does the following:
 - It creates an array of 10 double values between the range of 1.3 to 5.4 up to 2 floating points and prints out the array content. [2 marks]
 - It calculates the means of the generated arrays and prints out the results in 3 decimal points. [2 marks]

Here is a sample output:

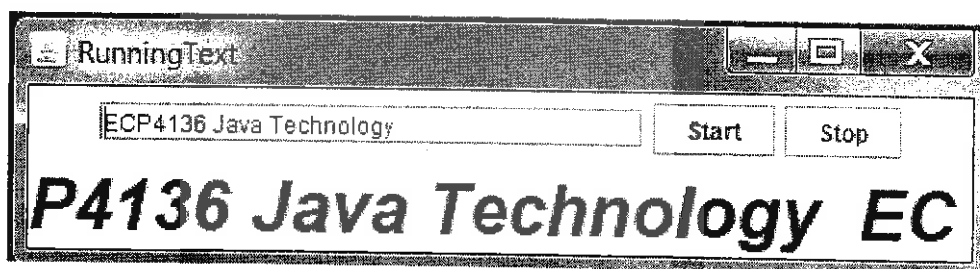
```
The generated values are [5.3, 2.01, 2.58, 4.27, 5.05, 1.51, 4.97,
2.63, 2.56, 3.26].
The arithmetic mean is 3.414
The geometric mean is 3.155
The quadratic mean is 3.655
```

Question 2

Define a **RunningText** class which inherits from **JPanel** with the following specifications:

- It has instance variables such as: [4 marks]
 - entryField**: **TextField** with size 25.
 - startButton**: **Button** with text "Start"
 - stopButton**: **Button** with text "Stop"
 - runningLabel**: **Label** which shows the running text.
 - index**: **int** which store the current displaying index position of the running text. Default value is 0.
 - timer**: **Timer** object which is initiated using an **ActionListener** object and 0.5s interval.
- When **startButton** is clicked, **timer** object will start and when **stopButton** is clicked, the **timer** object will stop. [2 marks]
- When **timer** elapses, it will extract the text from **entryField**, increment the **index** (without exceeding the text boundary) and use the index to cut and rejoin text with a space character before displaying it. [3 marks]
- It has **RunningText(text:String)** constructor that uses **text** as input for **entryField** and **runningLabel**, and attaches an **ActionListener** objects to the buttons. The font for **runningLabel** is Helvetica with bold and italic style and 36 font size. Use appropriate **LayoutManager** that will put field and buttons on the north and the labels in the centre. [6 marks]

This is a sample output:



Note: the **index** is at position 2.

Question 3

Create an abstract **Shape** class with the following specifications: [6 marks]

- It has **-shapeName:String** variable that holds the name of the shape.
- It has a **Shape(name:String)** constructor that receives the shape name.
- It declares the following abstract functions: **+getArea():double** and **+getPerimeter():double**.
- It has a **+toString():String** that returns the name, area and perimeter of the shape.

A regular polygon is an n-sided polygon in which all sides are the same length and all angles have the same degree. Create a **RegularPolygon** class that inherits from **Shape** with the following specifications:

- It has variables such as: [3 marks]
 - **-nSide:int** that define number of side. Default value is 3.
 - **-lSide:double** that define length of side. Default value is 1.0.
 - **-x:int** and **-y:int** that define the centroid. Default coordinate is (0,0).

- It provides accessors and mutators for all four variables. [4 marks]

- It has **RegularPolygon(n:int, l:double)** and **RegularPolygon(n:int, l:double, x:int, y:int)** constructors that receive and initialize the number of sides, the length of side and the coordinate. [4 marks]

- It implements both abstract functions. Area of a regular polygon can be calculated using the following formula $\frac{n \times l^2}{4 \times \tan\left(\frac{\pi}{n}\right)}$, where n is the number of side and l is

the length of the side. For example if the shape is a square with length 10 (then $n=4$, $l=10$), the perimeter and area are, 40 and 100. [3 marks]

End of Paper